Image-based walkthroughs from incremental and partial scene reconstructions

Kumar Srijan[1]
kumar.srijan@research.iiit.ac.in
Syed Ahsan Ishtiaque[1]
syed.ahsan.ishtiaque@gmail.com
Sudipta N. Sinha[2]
sudipta.sinha@microsoft.com
C.V. Jawahar[1]
jawahar@iiit.ac.in

[1] Center for Visual Information Technology,
IIIT Hyderabad, India
http://cvit.iiit.ac.in

[2] Microsoft Research
Redmond,USA
http://www.research.microsoft.com

Abstract

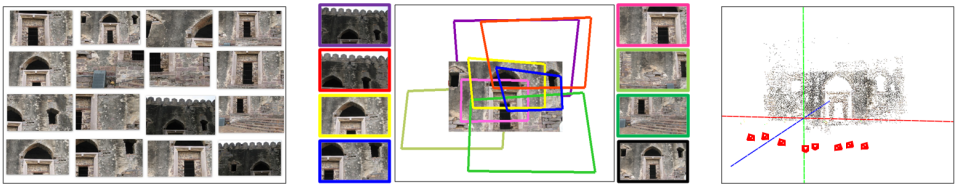
We present a scalable and incremental approach for creating interactive image-based walkthroughs from a dynamically growing collection of photographs of a scene. Prior approaches, such as [16], perform a global scene reconstruction as they require the knowledge of all the camera poses. These are recovered via batch processing involving pairwise image matching and structure from motion (Sfm), on collections of photographs. Both steps can become computational bottlenecks for large image collections. Instead of computing a global reconstruction and all the camera poses, our system utilizes several partial reconstructions, each of which is computed from only a small subset of overlapping images. These subsets are efficiently determined using a *Bag of Words*-based matching technique. Our framework easily allows an incoming stream of new photographs to be incrementally inserted into an existing reconstruction. We demonstrate that an image-based rendering framework based on only partial scene reconstructions can be used to navigate large collections containing thousands of images without sacrificing the navigation experience. As our system is designed for incremental construction from a stream of photographs, it is well suited for processing the ever-growing photo collections.

1 Introduction

The increasing popularity of digital photography and online photo-sharing sites such as Flickr is creating photo collections of landmarks and popular destinations around the world that are growing by the day. These massive datasets are visually interesting as they often capture a landmark site from a variety of viewpoints and in different illuminations and compositions. However, browsing such a massive unstructured photo collection can be difficult without any cues that indicate the relationship between the images. This has motivated a variety of approaches that try to organize photographs using geographical data, annotations, tags, etc. [6, 11, 16]. Attempts were also made to provide interactive and intuitive means of exploring photos and videos. The World-Wide Media Exchange (WWMX) [19] arranged

images on an interactive 2D map, PhotoCompas [8] clustered images based on time and location, Realityflythrough [7] explored video from camcorders instrumented with GPS and tilt sensors, and Kadobayashi and Tanaka [5] presented an interface for retrieving images using proximity to a virtual camera. Image-based walkthroughs which worked on the principles of image based rendering and virtual view synthesis had also been created but from controlled acquisition of imagery [1]. Aspen Movie Map allowed a user to take a virtual tour of the city of Aspen, Colarado by registering images captured from a moving car onto an interactive street map of the city. Google Street View and EveryScape provide panoramic views from various positions along many streets in the world. In Photowalker [18], a user can manually author a walkthrough of a scene by specifying transitions between pairs of images in a collection. In these systems, location is obtained from GPS or is manually specified. Johansson and Cipolla [4] developed a system where a user can take a photograph, upload it to a server where it is compared to an image database to receive location information.

Recent advances in computer vision in robustly solving image matching and the recovery of 3D structure and camera pose from images via *structure from motion* (Sfm), was exploited by the system dubbed *Photo-Tourism*[16]. It created extremely effective virtual 3D walkthroughs of a scene from unstructured Internet photo collections of popular tourist locations. The image correspondences and 3D camera poses recovered automatically by this system[16], made it possible for users to interactively navigate images registered in 3D. However, the underlying pipeline did not scale to large photo collections. Running times reported in[17] varied from 11 hours for a 1K image dataset to > 50 days for 8000 images. The primary computational bottlenecks were in the pairwise image matching step and the subsequent incremental 3D reconstruction stage where multiple rounds of global non-linear optimization referred to as *bundle adjustment* are performed. This is important since one of the overall goal in this system is to recover a single globally consistent reconstruction of the scene and all the cameras.



(a) (b) (c)

Figure 1: (a) An input image collection (b) Our interactive image navigation interface. (c) One of the multiple partial reconstructions of the scene, computed from the images shown in (b).

In this paper, we present a new system that generates interactive navigation of a photo-collection similar to Photo-tourism[16]. Our system however does not require a global 3D reconstruction of the scene and all the cameras. Rather, it relies only on partial local reconstructions which are independently estimated from subsets of nearby overlapping images and thereby allowing it to scale to large datasets.

In an image-based walkthrough, users primarily observe images or transition between image pairs at any time. Rendering a realistic transition between an image pair via image based rendering techniques [13] requires the knowledge of relative pose between the corresponding cameras, a sparse 3D reconstruction of points observed by these two cameras and

optionally a geometric reconstruction of the scene (required by advanced IBR techniques such as [13]). Therefore, a global Sfm reconstruction would allow direct transitions between any image pairs, as the relative pose of all pairs can be obtained from the global reconstruction. However, in practice only transitions between images whose views overlap tend to be the most useful. Our system limits the possible images that one would view next to a small set of proximal images whose views overlap with the current image. This is determined by the size of the subset for which a partial reconstruction is estimated.

Our approach for detecting these image subsets builds upon a state of the art image based retrieval technique [2, 9, 10, 14] which can efficiently retrieve duplicates or similar images based on visual appearance. This approach can be made scalable and more efficient by the use of inverted indices which map individual visual words to a list of images in which they occur. This can be very useful because generally only a small percentage of the visual words are present in an image. Each of the subsets are processed through a standard structure from motion pipeline. Restricting the size of the image subsets and processing them independent of each other reduces computation time and also makes it possible to exploit parallelism during the reconstruction stage. We demonstrate the ability to scale to large datasets without sacrificing much on the user's navigation experience.

The pairwise image matching bottleneck is addressed in a novel way by the work of [6], where *iconic* images are first recovered using a global scene descriptor (Gist features) for clustering the images into small collections. Then the expensive pairwise matching is applied within these small clusters to find the iconic image of each cluster as the image having the greatest number of features in common with rest of the images. Each of the iconic images are then verified with respect to its top n matches among the iconic images recovered using a visual word vocabulary based search. Another recent approach [11] avoids a globally consistent reconstruction of the scene in the context of robotic navigation to do scalable localization and mapping (to enable appearance based navigation) of the scene.

One of the key assumptions made by most previous systems is that all images will be available prior to processing, hence they are designed to process all the images in batch mode. However, online photo collections of important landmarks are often growing continuously and this indicates the need for an efficient online system that can incrementally insert new photographs into an existing reconstruction as they become available. Our incremental reconstruction framework maintains the photographs as a graph whose topology changes dynamically as new photographs become available. When a new image is successfully matched to existing photos in a pre-computed dataset, a new partial reconstruction is potentially added.

Our system is mostly immune to the various difficulties in computing a full global reconstruction via an incremental Sfm reconstruction approach, in particular its sensitivity to the choice of the initial image pair. We solve independent local Sfm problems and relax the need for global consistency in our reconstructions and camera poses. Thus, we avoid the catastrophic errors that occur when inaccuracies in camera pose estimation propagate and get compounded further along the sequence as new images, whose views overlap with the camera with erroneous pose, get added. This can lead to severe errors in large sections of a reconstruction [16, 17].

2 System Overview

Our system takes a set, (S), of uncalibrated images. The overall system can be broken up into stages in which operations are performed on this set of images. The relevant information required in the next stage, or for the incremental addition of images, is stored in the form of a graph at each stage. The following is the summary of the various stages in our system:

- **Obtaining Putative matches:** Use Vocabulary based techniques, for every image i $\in$S,$to efficiently and scalably find these to f neighbours$,N$_i$, as the images with similarity in appearance with image i. Create a directed graph of images, G_1, to store N_i for each i. Also store the histogram representation of the visual words present in i in the node corresponding to image i to allow easy comparison with a new incoming image during incremental insertion.

- **Geometric verification of the putative matches:** Estimate pairwise epipolar geometry to determine which images in N_i were viewing a common 3D structure as image i, for every $i \in S$. The verified set of images are called verified neighbors of i, V_i. The information about the verified neighbours is stored in the form of a directed graph of images, G_2.

- **Calibrating each image with respect to its true neighbors:** Use a Sfm system on the set $i \cup V_i$, for every $i \in S$, to find the parameters required for displaying images and making a transition from one image to another in a virtual setting while navigating though the scene. Store this information in the form of a directed graph of images, G_3, where every edge stores the parameters required for making a transition.

- **Creating mirrored edges to improve connectivity:** For every pair of images, add an edge from i to j, if an edge from j to i is present in G_3 and mark it as a mirrored edge. Store this new graph as G_4. While browsing the photo collection, the mirrored edges are simulated using their corresponding true edges due to which they were created. The set of images to which an edge go from the node corresponding to image i is called as Registered neighbors of i, R_i.

- **Addition of new images:** Repeat the previous four steps for the new image while updating the corresponding graphs at each stage. Also add the new image to the set S to allow matching with images coming in future.

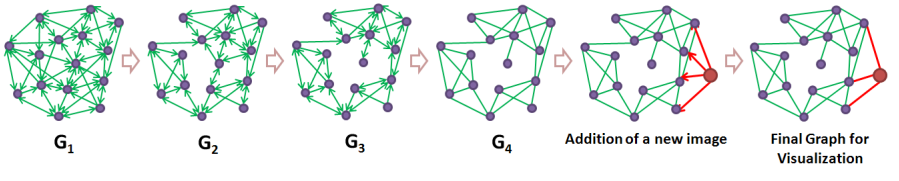


Figure 2: Overview of the system highlighting the process of insertion of a new image

The distribution of vertex degrees in graph G_4 indicates how connected it is. Even though the pairwise relations between the images are commutative, G_1, G_2 and G_3 are represented as directed graphs in order to bound the number of spatial verifications required and the number of images on which Sfm is performed for an image in one round. Provided the above two

points are true, it is safe to assume that under normal circumstances, a finite bound is put on the time taken in performing Sfm on the set of images $i \cup V_i$. This makes the time complexity of creating graph G_3 from G_2 approximately linear in the number of images and thus allowing it to scale to large number of images. Similarly, it is easy to see that creating G_2 from G_1 is also linear in the number of images in set S. Once a vocabulary is determined, then the representation of an image in the form a histogram of visual words depends upon the number of visual words (which is bounded) and also on the number of features extracted from every image (which is also bounded). Therefore, we need to do an $O(N^2)$ matching of histograms of the images. These histograms are sparse in general. In such a scenario, the determination of the top matches becomes $O(N)$ with respect to the number of images under consideration by the use of inverted indices of visual words as in [10].

Our visualization scheme allows the user to navigate the edges and nodes present in the graph G_4 in a virtual 3D world. Our visualization scheme is similar to that of [16] in terms of using a single proxy plane for view interpolation, but better viewpoint interpolations can also be computed using 3D geometric proxies, [13]. When the user is on a node corresponding to an image i, we show i in the center along with images in R_i which are shown as wireframes oriented in space in a way so as to provide a cue about the relationship among the images in the set $i \cup R_i$.

In the following sections, we describe our system in more detail.

3 Image Matching

Our goal is to recover for each image i, a partial scene reconstruction based on the currently available images which were directly matched to i. The relative pose of all the cameras and a sparse set of 3D points reconstructed from these images are represented in a local coordinate system. Putative matches for any image can be easily computed by matching its features with the features extracted from the rest of the images and ranking the images on the basis of number of feature matched with the query image. Quantization of features, by the use of visual word vocabularies, can be employed to speed up the process of matching the features. This visual word vocabulary used for this can be created from the features of a representative set of images of the whole dataset. As a result, an image can now be concisely represented as a histogram of visual word frequency and thereby reducing the problem of comparing images to comparing histograms. Alternatively, a Vocabulary tree based image search used in [3] can also be employed to find the top matching images of a query image.

The Bag of visual word based model ignores the position of the features, and hence, some of the features may get incorrectly matched *based on appearance* across images. If two images are looking at the same portion of the scene, then the geometric model predicted by the feature matches across the images is expected to accurately predict the position of most of the matching features across the images. Thus, verification of a matching pair can be done by examining the number of correct predictions the best geometric model, estimated from the feature matches, is able to do. Alternatively, these false matching images can be identified and removed by the Sfm system, i.e. creating G_3 directly from G_1, but we prefer to run the Sfm system only on the verified matches of an image as indicated in G_2. This saves time in case the Sfm system does an exhaustive pairwise matching of images. Additionally, it saves time for the bundle-adjustment step typically employed by Sfm systems to refine the estimates and further reject non matching images. This pre-verification can also be used in the identification of a good initial pair with respect to image i prior to applying Sfm on the set $i \cup V_i$. Specifying the initial pair for reconstruction as such increases the chance of image

i getting registered by the Sfm system when it is run on Image i and its neighbours. The following subsections give the implementation details:

3.1 Obtaining Putative Matches

As a first step towards identifying the verified neighbours of every image in S, we try to find the images which are similar in appearance. For this we extract robust SIFT features from all the images to apply a visual word vocabulary based image matching system. We used a representative set of images of the monument to create a context specific vocabulary. The SIFT features from these images were clustered using the Kmeans algorithm implemented on GPU. Alternatively, to avoid the overhead of creating a vocabulary, we also used the Vocabulary tree based image search used in [3] in some experiments. While creating the histograms of the images, we used Term Frequency(tf) to normalize the difference in the number of SIFT points extracted per image. We also used Inverse Document Frequency(idf) to downplay the importance of commonly occurring words. We measure the similarity between two images on the basis the similarity of the histogram of the two images measured using Cosine similarity distance function, used in many document retrieval techniques. The top 10 matches according to this score are recored for each image in G_1.

3.2 Geometric Verification

We do a refinement of the top matches returned for each image by the previous step to remove spurious matches. To verify an image pair, we first estimate a Fundamental matrix using RANSAC which can best explain the epipolar geometry of the matching features between the images. The inliers with respect to this fundamental matrix are computed and if the number of inliers is greater than some threshold (40 in our case), then the match is accepted. These images can be sent per se to the Sfm system for Geometry computation where some of the images may fail to get registered. The success of Geometry computation step depends significantly upon the initial pair of matching images used for starting the calibration process. The initial pair of images should have a good number of matching features and also have a good baseline. To identify the suitable matching pair for the reconstruction, we score them on the ratio of area which is covered by the inliers in each of the images as compared to the total area of the images. The area covered by the inliers is computed as the area covered by the convex hull enclosing all the inliers. The matching image which has the highest score with respect the concerned image, along with the concerned image, is taken as one of images in the initial pair for the reconstruction.

4 Generating Partial Reconstructions

After the spatial verification stage, we obtain a set of 2D correspondences within the set of images $i \cup V_i$. We now perform structure from motion on these images, to estimate a partial metric reconstruction of cameras and 3D points. We use BUNDLER [15], a freely available Sfm implementation that first generates an initialization for all cameras and points using an incremental seed and grow approach and then performs several rounds of bundle adjustment and outlier removal to refine the full camera calibration parameters and the sparse 3D points. Note that these camera pose estimates are with respect to a local coordinate frame, selected arbitrarily for each partial Sfm problem we solve. However, this is sufficient to recover the relative pose between camera i and any of its immediate neighbors in V_i.

The partial reconstruction corresponding to each image i is referred to as P_i. P_i comprises of reconstructed cameras for the set of images $i \cup V_i$ and a set of 3D points visible in these images. The relationship of any image i, with its registered neighbors R_i can be represented in the form of a directed graph, G_3, in which a directed edge is present from image i to every image $j \in R_i$. Any such edge means that a transition from the source to the destination image is possible while navigating the scene. Also, an edge from i to j can potentially be used to create an edge from j to i if it is not present. This is done by using the P_i as a proxy for P_j by centering P_i with respect to j while showing a transition for j to i. This makes the graph used for navigating the scene, G_4, undirected.

4.1 Image-based rendering framework

Our scene navigation scheme is inspired from the one used in the Phototourism system [16]. Initially, we show a 3D world corresponding to a partial reconstruction, P_i, of one of the images, i. Images are displayed by using a best fit plane(computed using RANSAC) corresponding of the points visible in the image as a proxy surface for projection. Initially, the virtual camera is placed congruent to the camera parameters recovered for image i and image i is projected on it proxy surface. Images in R_i are shown as wireframes of their corresponding projections on their proxy planes. The user can point and click at any of the wireframes to move to the partial reconstruction corresponding the a new image, j. This is done by showing a transition within P_i during which the virtual camera moves from the camera parameters corresponding to image i to the camera parameters corresponding to image j. Note that the recovery of metric camera calibration allows the possibility of estimating dense depth maps from the images, thus making it possible to use advanced image-based rendering techniques such as [12] for generating better transitions between photographs. The transition is accompanied by showing a fading in of image j and corresponding fading out of image i. At the end of the transition, we show the partial reconstruction P_j in the same way as described above. Thus, the user is able to navigate the scene using partial reconstructions.

5 Results

We have tested our system on a large collection of 6000 photographs of a heritage site, which we refer to as the FORT dataset. Some small subsets of this dataset which have been used

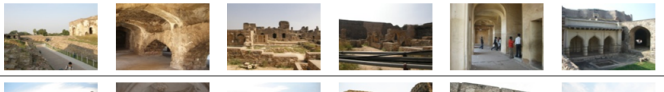
Name	Sample Images	N	T
Hilltop		114	42 minutes
Gate		135	1 hour
Courtyard		687	9 hours
Fort		5989	124 hours

Figure 3: Descriptions of HILLTOP, GATE, COURTYARD and FORT datasets used in our experiments. N shows the number of images and T shows the time taken by our system.

in specific experiments are the 135 images of the GATE dataset, and the HILLTOP dataset containing 114 images. We also tested the performance of our incremental system on the COURTYARD dataset containing 687 images taken under different lighting conditions.

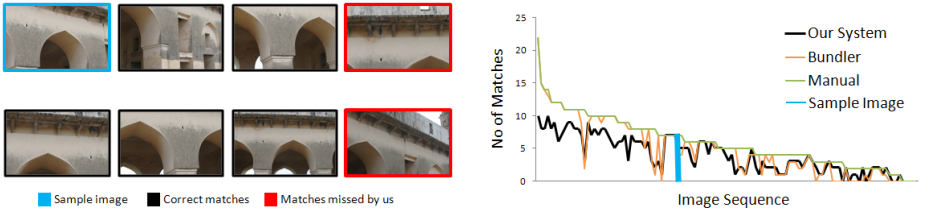


Figure 4: [left]Matches determined by our system from G_2 for a sample image (shown in blue) compared with its matches in G_m; [right]Comparison of the number of matches with the manual graph by the two systems for the HILLTOP DATASET

5.1 Experiments

We run the first 2 stages of the pipeline on the HILLTOP dataset to determine whether our system is able to robustly identify the correct neighbours of every image even when we are considering the top n matches given by a non geometric test (recorded in G_1) which are further refined by a simple RANSAC based geometric test (recorded in G_2). The vocabulary used for creating G_1 was created using a set of 1088 randomly sampled images of the whole monument. We created an undirected match graph(G_m), to be used as a ground truth, by manually comparing every image to every other image in the dataset. We also computed a match graph(G_b), in a manner similar to [16] by running BUNDLER [15] individually on the 3 different clusters present in the dataset. Figure 4[right] reports the number of matches of each image obtained from G_2 and G_b when compared against the set of neighbours obtained from G_m. Figure 4[left] shows an example image along with its matches obtained from G_2 compared with matches obtained from G_b.

In another experiment, we simulated a scenario in which we initialize with a small set of random images, and images arrive over time. For this we generate random permutations of the images in the GATE dataset. We initialize our system from the first 50 images. We use the

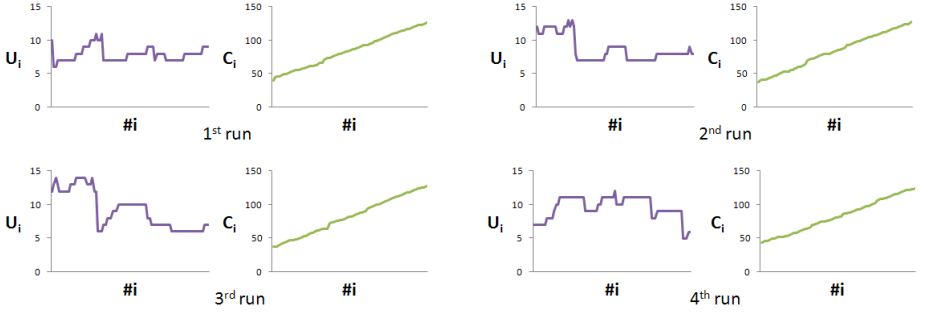


Figure 5: U_i represents the number of unregistered images and C_i represents the size of largest cluster for a set with i images; the figure shows that U_i converge to a small value and C_i grows continuously indicating that more and more images get registered to form a single cluster and the number of unregistered images decrease

pre-trained vocabulary tree generated from large number of images from the internet which was made available by [2] for computing G_1 and incrementally add the rest of the 85 images. This was repeated for other permutations. The images which do not have any neighbour in G_4 are marked as unregistered images and the size of the largest cluster is also shown (Figure 5) with the addition of each image. The graph G_4 is very fragmented in the beginning, but as more images get added, smaller disconnected components in the match graph get merged resulting in a largest cluster size of 126, 127, 128 and 124 images respectively for the four runs. The final match graph is well connected and provides a pleasant navigation experience. In a similar experiment with the COURTYARD dataset, we were able to get a cluster of 674 images out of 687 images while initializing from 200 images. This shows the applicability of our system on datasets of various sizes.

No of Images	Time taken by our system				Time taken by Bundler		
	G_1	G_2	G_3	Total	Matching	Bundle Adjustment	Total
55	22	401	541	**964**	719	310	**1029**
65	27	474	635	**1136**	1092	409	**1501**
75	31	563	738	**1332**	1427	563	**1990**
85	35	649	858	**1542**	1858	660	**2518**
95	39	785	1165	**1989**	2437	1023	**3460**
105	43	847	1437	**2327**	2914	1183	**4097**
115	47	975	1768	**2790**	3621	1470	**5091**
125	51	1079	2112	**3242**	4459	2146	**6605**
135	55	1164	2382	**3601**	4955	2487	**7442**

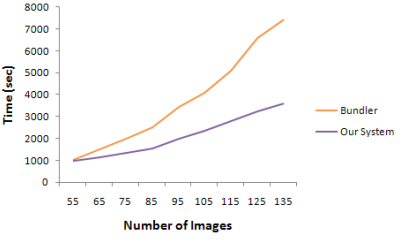
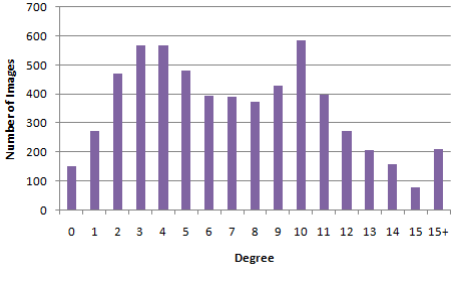


Figure 6: [left]Time taken in various stages of creating a walkthrough using our system as compared to time required to do a global reconstruction using BUNDLER [15] on the same set; [right] Graph comparing total time taken by the two systems.

In another experiment with the GATE dataset, we compare the time complexity of creating a walkthrough using our system to that of time taken to run the Sfm system on the whole dataset. We tested with sets of sizes 55 to 135 images with increments of 10 images. We use the pre-trained vocabulary tree made available by [2] for computing G_1. Time taken by our system is reported as the time required for matching (G_1), geometric verification (G_2) and creating partial scene reconstruction (G_3) for each image. The results are shown in 6[left].



Size of connected component	Number of Instances
1 - 10	277
11 - 20	17
21 - 30	5
31 - 100	2
101 - 500	2
501 - 1000	0
1000+	1

Figure 7: [left]Histogram showing the number of images of each degree in graph G_4 for the FORT dataset; [right] Table showing number of connected components of various sizes present in G_4 after running our system on the FORT dataset

In another experiment we demonstrate the scalability of our system on FORT dataset. We computed the graph G_1 of the set using the scene specific vocabulary created using a

representative set of 1088 images from the FORT dataset. We report the degree of each image in 7[left]. The average degree obtained per node is 7.1 for the registered images in graph G_4 even when we consider only 10 matches per image in G_1. The largest connected component we obtain is of 4249 images and another cluster of 453 images. Thus, we demonstrate that a good navigation experience can be built in a scalable fashion from our system as we are able to obtain clusters of decent size with good connectivity on large datasets.

6 Conclusions

We demonstrated that it is possible to achieve an image based browsing experience comparable to the one generated by a full reconstruction even by employing several partial reconstructions of a scene. The proposed approach is incremental, approximately linear in the number of images and massively parallel at every stage and hence easily scalable to very large image collections. The ability to incrementally grow our reconstruction makes it well suited for browsing the ever growing photo collections.

7 Acknowledgement

We would like to thank Noah Snavely for making BUNDLER [15] freely available for research purpouses and for providing necessary help on its proper usage.

References

[1] Daniel G. Aliaga, Thomas Funkhouser, Dimah Yanovsky, and Ingrid Carlbom. Sea of images. *Visualization Conference, IEEE*, 2002.

[2] Friedrich Fraundorfer, Christopher Engels, and David Nistér. Topological mapping, localization and navigation using image collections. In *IROS*, pages 3872–3877, 2007.

[3] Friedrich Fraundorfer, Changchang Wu, Jan-Michael Frahm, and Marc Pollefeys. Visual word based location recognition in 3d models using distance augmented weighting. In *3DPVT*, 2008.

[4] Bjorn Johansson and Roberto Cipolla. A system for automatic pose-estimation from a single image in a city scene. In *IASTED int. conf. Signal Processing, Pattern Recognition, and Applications*, 2002.

[5] Rieko Kadobayashi and Katsumi Tanaka. 3D viewpoint-based photo search and information browsing. In *SIGIR*, pages 621–622, 2005.

[6] Xiaowei Li, Changchang Wu, Christopher Zach, Svetlana Lazebnik, and Jan-Michael Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *ECCV (1)*, pages 427–440, 2008.

[7] Neil J. McCurdy and William G. Griswold. A systems architecture for ubiquitous video. In *MobiSys*, pages 1–14, 2005.

[8] Mor Naaman, Yee Jiun Song, Andreas Paepcke, and Hector Garcia-Molina. Automatic organization for digital photographs with geographic coordinates. In *JCDL*, pages 53–62, 2004.

[9] David Nistér and Henrik Stewénius. Scalable recognition with a vocabulary tree. In *CVPR (2)*, pages 2161–2168, 2006.

[10] James Philbin and Andrew Zisserman. Object mining using a matching graph on very large image collections. In *ICVGIP*, pages 738–745, 2008.

[11] Sinisa Segvic, Anthony Remazeilles, Albert Diosi, and François Chaumette. Large scale vision-based navigation without an accurate global reconstruction. In *CVPR*, 2007.

[12] Sudipta N. Sinha, Philippos Mordohai, and Marc Pollefeys. Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *ICCV*, 2007.

[13] Sudipta N. Sinha, Drew Steedly, and Richard Szeliski. Piecewise planar stereo for image-based rendering. In *ICCV*, 2009.

[14] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, 2003.

[15] Noah Snavely. Bundler, 2007. http://phototour.cs.washington.edu/bundler/.

[16] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3D. *ACM Trans. Graph.*, 25(3):835–846, 2006.

[17] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[18] Hiroya Tanaka, Masatoshi Arikawa, and Ryosuke Shibasaki. A 3D photo collage system for spatial navigations. In *Digital Cities*, pages 305–316, 2001.

[19] Kentaro Toyama, Ron Logan, and Asta Roseway. Geographic location tags on digital images. In *ACM Multimedia*, pages 156–166, 2003.